PROJECT

SUBMITTED TO:
DR MUKHTIAR BANO

SUBMITTED BY:
TANZEELA ASGHAR (032)
LAIBA SOHAIL (016)
BISMA ALI (008)

BSE (V-A)

PLATFORM AS A SERVICE:

1-PAAS IN THE CONTEXT OF CLOUD COMPUTING:

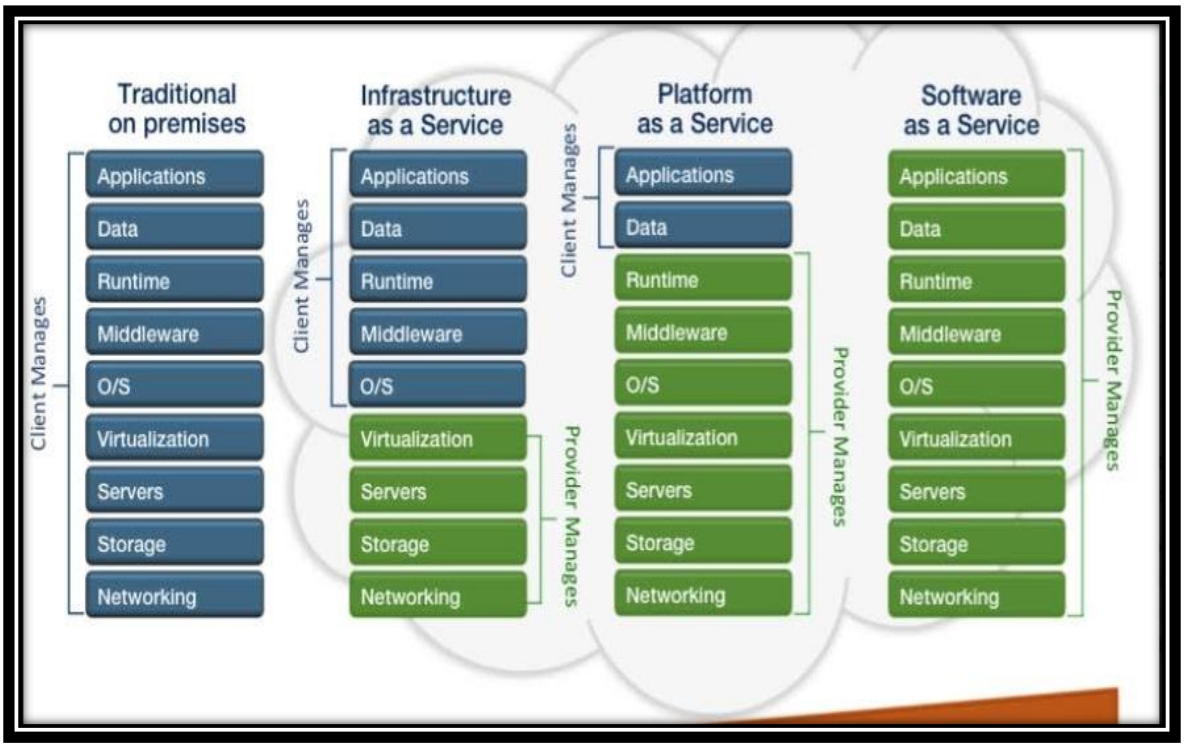
Cloud computing is a paradigm offering scalable and shareable physical and virtual resources through network access. It encompasses infrastructure, platform, and application capabilities. Platform as a Service (PaaS) specifically focuses on developing, deploying, and operating customer applications, distinguishing itself from other capabilities like processing and storage.

2-TRADITIONAL ON-PREMISES:

In traditional on-premises deployment, customers are responsible for acquiring, configuring, and operating the entire application ecosystem, including hardware, software stack, databases, and management tools. This involves handling servers, data storage, networks, operating systems, middleware, runtimes, custom code, and deployment/update processes.

3-PLATFORM AS A SERVICE (PAAS):

Platform-as-a-Service (PaaS) primarily caters to application developers, offering a cloud service rendering of application infrastructure, including middleware capabilities. PaaS handles installation, configuration, and operation of the application infrastructure, allowing customers to focus solely on their application code. It provides a diverse set of services and APIs, enabling immediate productivity for developers without requiring specialized skills. PaaS differs from Software-as-a-Service (SaaS) by supporting the creation and use of custom application code, while contrasting with Infrastructure-as-a-Service (IaaS) by managing application middleware stacks, though it may blend features of both. PaaS facilitates building "born on the cloud" applications and offers flexibility in tailoring capabilities for specific business needs.



4-ENABLING TECHNIQUES:

4.1-VIRTUALIZATION:

Virtualization, at its core, abstracts the underlying physical infrastructure, allowing developers to deploy applications without concerning themselves with hardware specifics. This abstraction provides benefits such as resource isolation, elasticity, and multi-tenancy. It simplifies management and enhances security, ensuring applications can run on diverse hardware seamlessly.

4.2-MICROSERVICE ARCHITECTURE:

Microservices architecture is an approach where applications are composed of small, independent services. These services, focused on specific business capabilities, communicate through well-defined APIs. This architecture promotes agility by decomposing monolithic applications, enabling independent deployment, granular scalability, and enhanced resilience. PaaS platforms supporting microservices architecture empower developers to build, deploy, and scale services independently.

4.3-CONTAINERIZATION:

Containerization involves encapsulating applications and their dependencies into lightweight, portable containers. Containers ensure consistency across diverse environments, addressing deployment challenges. They offer portability, allowing applications to run consistently across different infrastructure and cloud providers. PaaS platforms leverage containerization for resource efficiency, scalability, and efficient orchestration using tools like Kubernetes.

5-CHARACTERISTICS OF PLATFORM-AS-A-SERVICE:

Major characteristics of PaaS include

- support for custom applications
- provision of runtime environments for various programming languages
- rapid deployment mechanisms
- a range of middleware capabilities
- services delivered via APIs
- preconfigured capabilities for simplicity
- API management and security features.

PaaS also provides tools for developers, operations capabilities, and supports porting existing applications. It is designed for "born on the cloud" applications, particularly Systems of Engagement, and can be used to build applications offered as Software as a Service (SaaS).

6- BENEFITS OF PLATFORM-AS-A-SERVICE

PaaS systems enable the realization of the benefits of cloud computing as a whole:

- Scalability including rapid allocation and deallocation of resources with a pay-as-you-use model (noting that the use of individual resources can vary greatly over the lifecycle of an application)
- Reduced capital expenditure
- Reduced lead times with on-demand availability of resources
- Self-service with reduced administration costs

- Reduced skill requirements
- Support of team collaboration
- Ability to add new users quickly

7-PAAS PROVIDERS:

Amazon Web Services (AWS) Elastic Beanstalk:

- AWS provides Elastic Beanstalk as a fully managed PaaS offering. It supports multiple programming languages and frameworks, making it easier to deploy and manage applications.

Microsoft Azure App Service:

- Azure App Service is Microsoft's PaaS offering that supports various programming languages, frameworks, and integration with Azure services. It allows developers to build, deploy, and scale applications with ease.

Google Cloud App Engine:

- Google Cloud's App Engine is a fully managed serverless platform that enables developers to build and deploy applications. It supports multiple languages and automatically scales based on demand.

IBM Cloud Foundry:

- IBM Cloud offers Cloud Foundry as a PaaS solution, providing an open-source platform that supports multiple programming languages. It allows developers to focus on writing code without worrying about the underlying infrastructure.

8-EXAMPLES OF PAAS:

- Heroku is a cloud platform that allows developers to build, deploy, and scale applications easily. It supports multiple programming languages and provides a simple and intuitive interface.
- IBM Cloud Foundry is an open-source PaaS solution that supports multiple programming languages. It allows developers to focus on writing code while abstracting away the complexities of infrastructure management.
- App Cloud by Salesforce is a PaaS platform focused on building applications for customer relationship management (CRM). It includes tools like Force.com for application development

9-UNDERSTAND PAAS END-TO-END APPLICATION ARCHITECTURE:

The "12-factor app" methodology, crucial for cloud-native applications, is supported by PaaS systems. Key concepts include maintaining one codebase with multiple deploys, explicit declaration of dependencies, storing configuration separately from the code, treating services as loosely coupled resources, and separating build, release, and run stages. Other factors involve stateless, horizontally scalable architecture, emphasising agility and test-driven development. Stateless applications use separate services for data, enabling scalability,

failover, and redundancy for resilience and high availability. This approach contrasts with monolithic architectures, where all components are packaged and deployed as a single entity.

References

[1] ISO/IEC (2014): Cloud Computing Reference Architecture.
http://standards.iso.org/ittf/PubliclyAvailableStandards/c060545_ISO_IEC_17789_2014.zip
[2] ISO/IEC (2014): Cloud Computing Overview and Vocabulary.
http://standards.iso.org/ittf/PubliclyAvailableStandards/c060544_ISO_IEC_17788_2014.zip
[3] Gartner (2014): Platform as a Service: Definition, Taxonomy and Vendor Landscape, 2014.
https://www.gartner.com/doc/2833022/platform-service-definition-taxonomy-vendor
[4] CSCC (2014): Practical Guide to Cloud Service Agreements V2.0, 2014.
http://www.cloud-council.org/CSCC_Practical_Guide_to_Cloud_Service_Agreements_Version_2.0.pdf
[5] CSCC (2014): Interoperability and Portability for Cloud Computing: A Guide, 2014.
http://www.cloud-council.org/CSCC-Cloud-Interoperability-and-Portability.pdf
[6] 12factor.net (2014): The 12 Factor App.
http://12factor.net/